CLAIMS

What is claimed is:

1. A system for assembling a data packet, the system comprising:

a first memory element for storing a sequence number and a sub-channel identifier for an incoming data packet;

a second memory element for storing one or more revised packet fragment;

a third memory element for storing one or more unrevised packet fragment;

a fourth memory element for storing a starting address, wherein the starting address may be the starting address of a revised packet fragment or the starting address of an unrevised packet fragment, wherein the first memory element identifies portions of the fourth memory element associated with the sequence number; and

one or more processors, wherein the one or more processors are configured to create a modified data packet by combining the one or more unrevised packet fragments and the one or more revised packet fragment, wherein the modified data packet is associated with the sequence number and the sub-channel identifier.

2. The system for assembling a data packet of claim 1, wherein the one or more processors create the modified data packet from 64-bit bursts.

3. The system for assembling a data packet of claim 1, wherein the one or more unrevised packet fragments are 64-bit words unaligned.

4. The system for assembling a data packet of claim 1, wherein the one or more revised packet fragments are 64-bit words aligned.

5. The system for assembling a data packet of claim 1, wherein the incoming data packet is a datagram.

6. The system for assembling a data packet of claim 1, wherein entries in the fourth memory element are organized as a bit validity field, a size of data field, and a location of data field.

7. A packet modification system comprising:

a buffer for providing some or all of a packet as packet portions and providing the portions in parallel over a first data path having a first width to a modification engine;

the modification engine configured to receive the packet portions over the first data path, and, responsive thereto, modify one or more of the packet portions; and

an assembler for assembling a packet from one or more modified and one or more unmodified packet portions, and providing the assembled packet over a second data path having a second width less than the first width.

8. The system of claim 7 wherein the buffer comprises a plurality of memories configured to store a packet and provide portions thereof in parallel over the first data path to the modification engine.

9. A packet modification system comprising:

a buffer for buffering the packet upon or after ingress thereof by the system;

a modification engine for modifying one or more portions of the packet; and

an assembler for assembling one or more modified portions of the packet with one or more unmodified portions of the packet as retrieved directly from the buffer to form an assembled packet on an egress path of the system.

10. The system of claim 9 wherein the modification engine is configured to provide a list indicating which portions of the assembled packet are to comprise modified portions of an ingress packet, and which portions are to comprise unmodified portions of the ingress packet, and the assembler is configured to assemble the assembled packet responsive to the list.

11. A packet modification system comprising:

means for buffering a packet upon or after ingress thereof to the system;

means for slicing some or all of a packet into portions and providing the portions in parallel over a first data path having a first width to modification means, the modification means configured for modifying the packet, or one or more portions thereof; and

means for assembling one or more modified portions of the packet with one or more unmodified portions of the packet retrieved directly from the means for

buffering to form an assembled packet, and providing the assembled packet over a second data path having a second width less than the first width.

12. A method of modifying a packet comprising:

a step for buffering a packet in a buffer upon or after ingress thereof;

a step for slicing some or all of a packet into portions and providing the portions in parallel over a first data path having a first width to a modification engine;

in the modification engine, a step for modifying the packet, or one or more portions thereof; and

a step for assembling one or more modified portions of the packet with one or more unmodified portions of the packet as directly retrieved from the buffer to form an assembled packet, and providing the assembled packet over a second data path having a second width less than the first width.

13. A system for assembling a data packet, the system comprising:

a first memory element, the first memory element including a plurality of buffers, wherein one of the plurality of first memory element buffers is associated with a first memory element buffer identifier and the one first memory element buffer is configured for storing a packet kill flag, an end-of-packet flag, a packet offset, a second memory element buffer length, a start-of-packet pointer, a sub-channel identifier, and a sequence number;

a second memory element, the second memory element including a plurality of buffers, wherein one of the plurality of second memory element buffers is associated with the first memory element buffer identifier and the one second memory element buffer is configured to include a valid bit, an indication of data size, the location of data;

an assembler, wherein the assembler aligns data forming an egress data packet, wherein the data is aligned using the packet offset; and

a processor, wherein the processor is configured to search the first memory element for a next sub-channel sequence number, wherein the processor is configured to control the assembler to create the egress data packet, wherein the egress packet is associated with the sub-channel.

14. The system for assembling a data packet of claim 13, wherein the processor is configured to create the egress data packet from the location of data when the valid bit is set and the data size is set to zero.

15. The system for assembling a data packet of claim 13, wherein the assembler aligns revised and unrevised data to create the egress data packet.

16. The system for assembling a data packet of claim 13, wherein the assembler aligns revised and unrevised data to create the egress data packet, wherein the revised data is revised pursuant to modification recipe associated with an ingress data packet.

17. The system for assembling a data packet of claim 13, wherein the second memory element is written in ascending byte order.

18. A method for assembling a data packet, the method comprising steps of:

storing a sequence number and a sub-channel identifier for an incoming data packet in a first memory element;

storing one or more revised packet fragment;

storing one or more unrevised packet fragment;

storing a starting address in a second memory element, wherein the starting address may be the starting address of a revised packet fragment or the starting address of an unrevised packet fragment, wherein the first memory element identifies portions of the second memory element associated with the sequence number; and

creating a modified data packet by combining the one or more unrevised packet fragments and the one or more revised packet fragment, wherein the modified data packet is associated with the sequence number and the sub-channel identifier.

19. The method for assembling a data packet of claim 18, wherein the step of creating a modified data packet includes creating the modified data packet from 64-bit bursts.

20. The method for assembling a data packet of claim 18, wherein the one or more unrevised packet fragments are 64-bit words unaligned.

21. The method for assembling a data packet of claim 18, wherein the one or more revised packet fragments are 64-bit words aligned.

22. The method for assembling a data packet of claim 18, wherein the incoming data packet is a datagram.

23. The method for assembling a data packet of claim 18, wherein entries in the second memory element are organized as a bit validity field, a size of data field, and a location of data field.

24. A system for assembling a data packet, the system comprising:

means for storing a sequence number and a sub-channel identifier for an incoming data packet in a first memory element;

means for storing one or more revised packet fragment;

means for storing one or more unrevised packet fragment;

means for storing a starting address in a second memory element, wherein the starting address may be the starting address of a revised packet fragment or the starting address of an unrevised packet fragment, wherein the first memory element identifies portions of the second memory element associated with the sequence number; and

means for creating a modified data packet by combining the one or more unrevised packet fragments and the one or more revised packet fragment, wherein the modified data packet is associated with the sequence number and the sub-channel identifier.

25. The system for assembling a data packet of claim 24, wherein the means for creating a modified data packet includes means for creating the modified data packet from 64-bit bursts.

26. The system for assembling a data packet of claim 24, wherein the one or more unrevised packet fragments are 64-bit words unaligned.

27. The system for assembling a data packet of claim 24, wherein the one or more revised packet fragments are 64-bit words aligned.

28. The system for assembling a data packet of claim 24, wherein the incoming data packet is a datagram.

29. The system for assembling a data packet of claim 24, wherein entries in the second memory element are organized as a bit validity field, a size of data field, and a location of data field.